



Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile



MINISTERO DELLO SVILUPPO ECONOMICO



Ricerca di Sistema elettrico

Integrazione di funzionalità e algoritmi per la localizzazione e la navigazione autonoma all'interno di un veicolo elettrico e relativa sperimentazione

E. Bellocchio, M. Leomanni, G. Costante, M.L. Fravolini, P. Valigi

A.D. 1308
unipg

UNIVERSITÀ DEGLI STUDI
DI PERUGIA



Report RdS/PTR2020/033

Integrazione di funzionalità e algoritmi per la localizzazione e la navigazione autonoma all'interno di un veicolo elettrico e relativa sperimentazione

E. Bellocchio, M. Leomanni, G. Costante, M.L. Fravolini, P. Valigi (Intelligent Systems, Automation and Robotics Laboratory (ISARLab) - Dipartimento di Ingegneria, Università degli Studi di Perugia)

Marzo 2022

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - II annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package: Local Energy District

Linee di attività: LA43 - Integrazione di funzionalità autonome in un veicolo elettrico

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "Integrazione di funzionalità autonome in un veicolo elettrico e interoperabilità con l'infrastruttura "smart road""

Responsabile scientifico ENEA: Sergio Taraglio

Responsabile scientifico Dipartimento di Ingegneria - Università degli Studi di Perugia: Mario Luca Fravolini

Responsabile tecnico Dipartimento di Ingegneria - Università degli Studi di Perugia: Gabriele Costante

Indice

SOMMARIO	5
1 INTRODUZIONE	6
2 NAVIGAZIONE AUTONOMA IN AMBIENTE URBANO	8
2.1 SISTEMA DI NAVIGAZIONE AUTONOMA	8
2.1.1 Modulo di localizzazione	9
2.1.2 Modulo di pianificazione della traiettoria	12
2.1.3 Modulo di generazione e controllo della traiettoria	13
2.1.4 Modulo di rilevazione degli ostacoli	14
2.1.5 Modulo di conversione di basso livello	15
3 ESPERIMENTI SU SEQUENZE RACCOLTE DAL VEICOLO IN AMBIENTE URBANO	16
4 ESPERIMENTI DI GUIDA AUTONOMA IN AMBIENTE URBANO	21
5 CONCLUSIONI	24
6 RIFERIMENTI BIBLIOGRAFICI	25
7 ABBREVIAZIONI ED ACRONIMI	26
8 BREVE CURRICULUM VITAE DEL GRUPPO DI LAVORO DEL DIPARTIMENTO DI INGEGNERIA, UNIVERSITÀ DEGLI STUDI DI PERUGIA	27
<i>Lista delle pubblicazioni recenti più rilevanti</i>	28

Sommario

Questo documento contiene il report delle attività svolte dal gruppo di ricerca ISARLab del Dipartimento di Ingegneria dell'Università degli Studi di Perugia nel periodo 01/01/2021 - 31/12/2021 relativamente al progetto *“Integrazione di funzionalità autonome in un veicolo elettrico e interoperabilità con l'infrastruttura “smart road”*”, di seguito indicato per brevità *“Smart Road”*.

Durante questo periodo di tempo le attività svolte sono state incentrate principalmente nello sviluppo e nella sperimentazione di un sistema di navigazione per la guida autonoma di un veicolo elettrico in ambiente urbano. Il sistema di guida si basa sull'analisi dei dati provenienti dai sensori presenti sul veicolo, ovvero telecamere, GPS, sensore inerziale e LIDAR.

Il sistema è stato poi testato su sequenze di dati raccolte dal veicolo, misurando le capacità di localizzazione e di stima della traiettoria.

La prima parte di questo documento contiene una descrizione dei dettagli tecnici ed implementativi del sistema realizzato, mentre la seconda parte contiene i risultati sperimentali ottenuti con le sequenze raccolte.

1 Introduzione

In questa ultima fase del progetto è stato realizzato il sistema di guida autonoma da installare sul veicolo elettrico e sono stati effettuati dei test di valutazione su sequenze raccolte dal veicolo durante la guida in strada. Le sequenze raccolte, contenenti tutte le misure dei sensori presenti sul veicolo (ovvero le scansioni laser effettuate tramite LIDAR, le coppie di immagini sincronizzate raccolte dalla stereo camera, le posizioni del veicolo stimate dal modulo GPS ed i valori delle grandezze inerziali fornite dalla IMU), sono state utilizzate in post-processing per valutare le capacità di localizzazione, di stima dell'odometria del veicolo e di realizzazione della mappa dell'ambiente. Grazie alle sequenze messe a disposizione dai colleghi del Laboratorio di Robotica ed Intelligenza Artificiale del Centro di Ricerca di Casaccia, è stato possibile valutare in un contesto reale le strategie di SLAM (Simultaneous Localization And Mapping) basati sull'utilizzo di sensore LIDAR ed i metodi di Visual Odometry e di SLAM visuale, basati quindi sull'utilizzo delle telecamere.

Inoltre, è stato sviluppato il modulo necessario per la conversione dei segnali di controllo "di alto livello" restituiti dal sistema di navigazione autonoma (rappresentati da una serie di valori di velocità lineari e di rotazione) in segnali "di basso livello" che tengano conto della cinematica del veicolo e che possano essere trasmessi direttamente ai microcontrollori collegati ai motori elettrici (segnali rappresentati principalmente da una serie di valori per l'angolo di sterzata e per la velocità di avanzamento).

Sensore	Modello
Telecamere	FLIR BlackFly in modalità stereocamera
LIDAR	Velodyne Puck Lite VLP16
IMU + GPS + Bussola	PX4 Autopilot

Tabella 1: panoramica dei sensori presenti sul veicolo elettrico ed utilizzati in fase di valutazione degli algoritmi di localizzazione

In tabella 1 sono mostrati i sensori installati sul veicolo elettrico utilizzati per la valutazione degli algoritmi di localizzazione. Le immagini sono state raccolte da una coppia di telecamere FLIR Blackfly configurate in modalità stereo-camera, in grado di raccogliere immagini a 10 FPS e ad una risoluzione di 720 X 580 pixel. Le scansioni LIDAR sono state raccolte da un modulo Velodyne puck lite VLP 16 ad una frequenza di 10 Hz, sensore in grado di rilevare oggetti ed ostacoli con un angolo di cattura di 360 gradi e ad una distanza di circa 90 metri. Le misure inerziali, le posizioni GPS ed i dati della bussola vengono raccolti da un modulo PX4 Autopilot. Lo sviluppo del software e la gestione dei dati forniti dai sensori sono stati condotti tramite gli strumenti messi a disposizione dal framework di sviluppo ROS (Robotic Operating

System). Infatti, il sistema di localizzazione e navigazione autonoma ed i moduli di interfacciamento con i sensori sono rappresentati da nodi ros, inoltre tutti dati dei sensori vengono resi disponibili al resto dell'ecosistema software tramite dei canali di trasmissione (chiamati topics) messi a disposizione dal framework ROS, ogni misura viene codificata tramite una struttura dati standardizzata, chiamato messaggio ROS.

2 Navigazione autonoma in ambiente urbano

L'ultima parte delle attività condotte ha avuto come scopo la realizzazione e la sperimentazione di un sistema di navigazione autonoma che utilizza i sensori presenti sul veicolo elettrico.

2.1 Sistema di navigazione autonoma.

È stato implementato un sistema software in grado di guidare il veicolo elettrico lungo un percorso stradale, evitando in modo autonomo ostacoli fissi e dinamici presenti lungo il percorso.

Il sistema di navigazione implementato è costituito da diversi blocchi funzionali, che hanno lo scopo di risolvere problematiche specifiche riguardanti la navigazione autonoma (es: interfacciamento dei sensori, elaborazione delle misure, gestione dei comandi, controllo dei motori, algoritmi di filtraggio, localizzazione, pianificazione del percorso, obstacle avoidance, ecc.). Tutti i moduli sono stati implementati utilizzando il framework di sviluppo ROS (Robot Operating System) ed ognuno di essi è rappresentato da uno o più nodi ROS. I moduli che compongono il sistema di navigazione autonoma sono i seguenti:

- **Modulo di localizzazione del veicolo:** permette di conoscere in ogni momento la posizione e l'orientamento del veicolo. Tale informazione viene utilizzata per il calcolo dei comandi di movimento, al fine di inseguire la traiettoria di riferimento fornita dai moduli di pianificazione.
- **Modulo di pianificazione della traiettoria:** permette di coordinare il percorso (o missione) che deve essere seguito dal veicolo. Tale modulo gestisce una lista di punti, chiamati *waypoints*, che dovranno essere raggiunti sequenzialmente, controllando anche l'istante temporale in cui il veicolo deve raggiungere un determinato waypoint.
- **Modulo di generazione e controllo della traiettoria:** permette di pianificare la traiettoria del veicolo tra un waypoint e l'altro ed effettuare il calcolo dei comandi da fornire alla piattaforma in base al percorso, alla cinematica e alle caratteristiche strutturali della piattaforma.
- **Modulo di rilevazione degli ostacoli:** Questo modulo ha lo scopo di rilevare gli ostacoli fissi o dinamici che si possono incontrare durante la navigazione e di

attuare delle procedure di *avoidance* per aggiornare la traiettoria al fine di evitare la collisione.

- **Modulo di conversione dei comandi per il controllo di basso livello:** permette di convertire i segnali forniti dal modulo di controllo della traiettoria, composti da valori di velocità lineare ed angolare, in segnali di basso livello, composti da valori di angolo di sterzata e velocità di avanzamento, da inviare alle schede di controllo collegate direttamente ai motori elettrici.

Nei paragrafi seguenti viene fornita una descrizione approfondita degli algoritmi e delle strategie utilizzate per implementare ciascuno dei moduli descritti.

2.1.1 Modulo di localizzazione

La localizzazione del veicolo può essere svolta da un algoritmo di stima dell'odometria (stima della traiettoria percorsa) o di localizzazione e mapping simultaneo dell'ambiente (SLAM- Simultaneous Localization and Mapping), o da più algoritmi, che vengono combinati insieme attraverso strategie di fusione sensoriale.

Gli algoritmi di localizzazione implementati si dividono principalmente secondo due modalità:

Localizzazione rispetto al sistema di riferimento del veicolo ("relativa"): La localizzazione del veicolo avviene tramite algoritmi SLAM (Simultaneous Localization and Mapping) o di stima dell'odometria. Negli approcci SLAM la piattaforma, processando le misure provenienti dai sensori, è in grado di creare una mappa dell'ambiente che lo circonda (non noto a priori) e di localizzarsi. Negli approcci di stima di odometria il veicolo, processando i dati, calcola in ogni istante la roto-traslazione rispetto al momento precedente. In ogni momento il veicolo è in grado di determinare la sua posizione rispetto all'origine rappresentata dal punto di inizio dell'esperimento. Sono stati testati diversi algoritmi di questa tipologia, in base al sensore principale utilizzato:

- **Algoritmi LIDAR-based:** algoritmi di localizzazione che sfruttano principalmente le scansioni di un sensore LIDAR (Light Detection And Ranging). Nei nostri esperimenti il sensore utilizzato è stato il Velodyne VLP-16, installato nel veicolo. Gli algoritmi testati sono stati:
 - **FLOAM [1]:** metodo presentato come evoluzione ottimizzata e più veloce di LOAM. Metodo di SLAM basato sull'elaborazione di dati provenienti da LIDAR rotativi (es. Velodyne).

- **SC-LEGO-LOAM:** metodo di SLAM LIDAR-based basato sulla strategia proposta in [2] come evoluzione del metodo LOAM [3] e che utilizza un algoritmo di loop closing basato su strategia Scan Context [4]. questo metodo può essere utilizzato solamente con il sensore LIDAR o in combinazione con un modulo di misura inerziale (IMU- Inertial Measurement Unit). In figura 1 viene mostrata la mappa ricavata dall'esecuzione di questo metodo sulla sequenza raccolta dal centro ENEA.
- **LIO-SAM [5]:** metodo SLAM LIDAR-based che prevede l'integrazione di una IMU per la stima precisa della posa del veicolo.

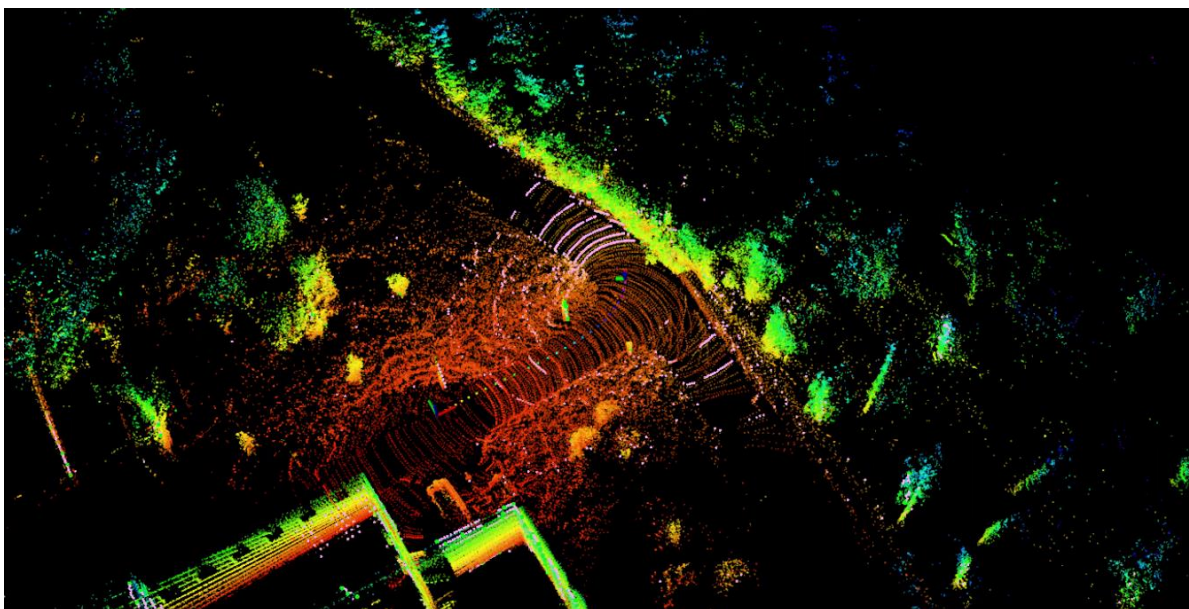


Figura 1: Mappa 3D ottenuta tramite misure LIDAR con algoritmo SC-LEGO-LOAM.

- **Algoritmi vision-based:** Algoritmi di localizzazione che sfruttano principalmente le immagini provenienti da telecamere. Nei nostri esperimenti è stata utilizzata la coppia di telecamere in configurazione stereo installate sul veicolo elettrico. Gli algoritmi testati sono stati:
 - **ORB-SLAM3 [6]:** metodo SLAM basato sull'analisi di immagini provenienti da telecamere monoculari o stereo con l'eventuale presenza di una IMU. La strategia di localizzazione e mapping si basa sull'estrazione di punti di interesse all'interno delle immagini (keypoints) descritte da feature ORB, di cui ne vengono ricercate le corrispondenze sia tra coppie di immagini (nel caso di utilizzo con stereo camera: matching), sia tra immagini scattate in istanti di tempo differenti

(tracking). Dalle corrispondenze ottenute si cerca di costruire una mappa metrica rappresentante l'ambiente in cui si muove il veicolo. La mappa calcolata viene infine utilizzata per la localizzazione. Per una migliore stima della posizione e dell'orientamento del veicolo è anche possibile utilizzare le misure fornite da una IMU. In figura 2 viene mostrato il funzionamento del metodo in modalità mono-camera.

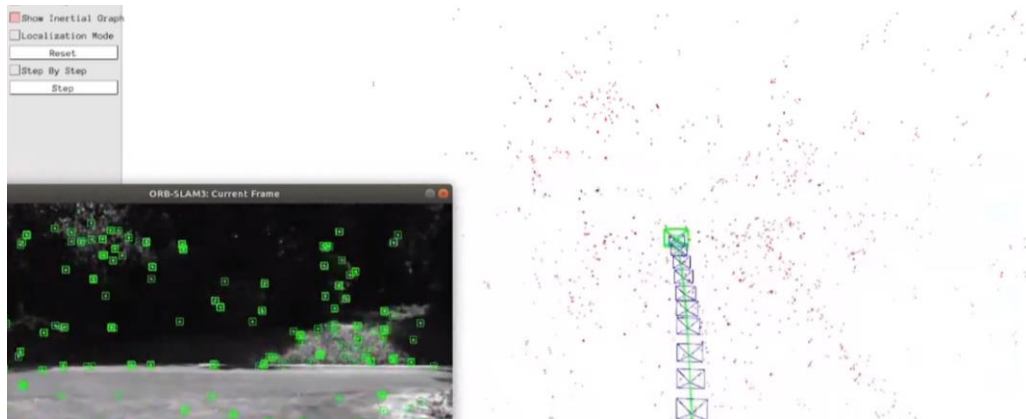


Figura 2: traiettoria del robot e mappa dell'ambiente calcolata da ORB-SLAM3 (destra) ed esempio di fotogramma elaborato con indicati i keypoints estratti (sinistra).

- **DSO [7]:** Metodo di visual odometry di tipo diretto, ossia che si basa sull'utilizzo diretto dei pixel, senza la loro rappresentazione tramite features visuali. La stima dell'odometria avviene tramite la minimizzazione dell'errore fotometrico tra un frame e l'altro e tramite la stima della profondità dei pixel. Il metodo risulta inoltre sparso, in quanto non utilizza tutti i pixel delle immagini ma soltanto un loro sottoinsieme. Il metodo può essere utilizzato sia in modalità monoculare che con l'utilizzo della stereo-camera. In figura 3 è mostrato un esempio di esecuzione di DSO.

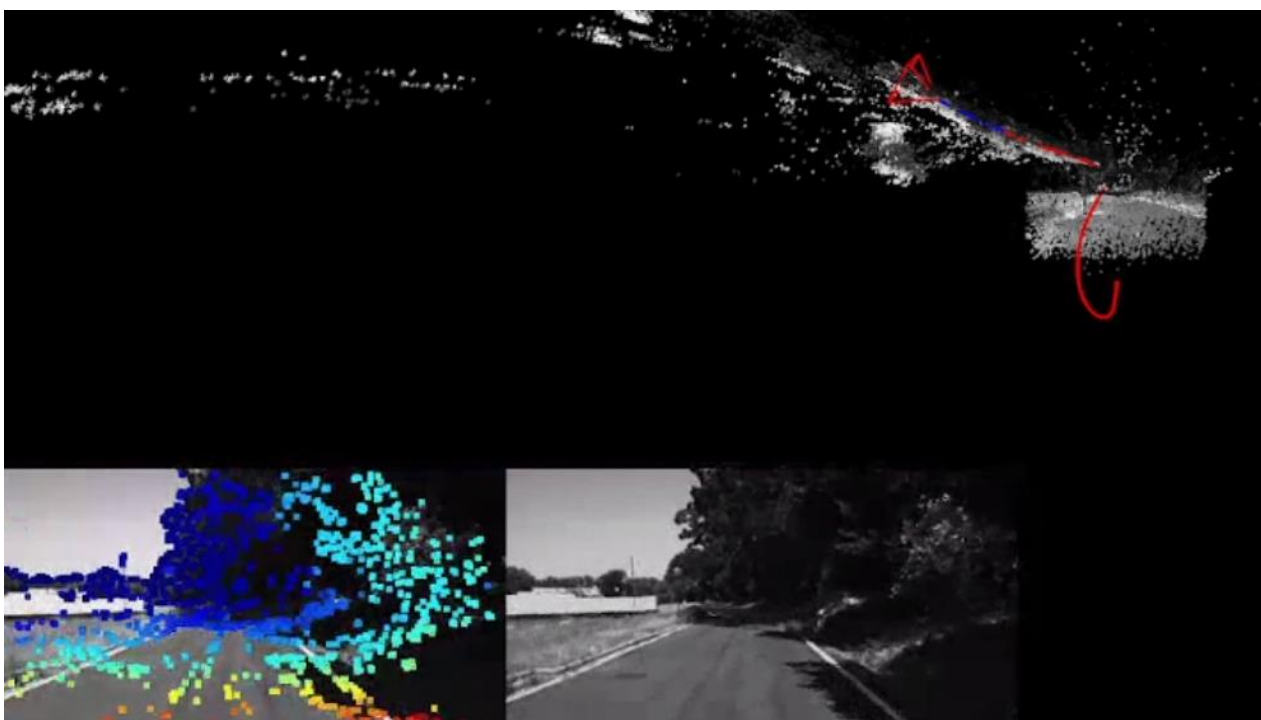


Figura 3: Traiettorie del veicolo ricostruite da DSO (in alto) e le immagini elaborate dal metodo (in basso).

Localizzazione rispetto al sistema di riferimento GPS (“assoluta”): In questo caso, la localizzazione del veicolo è effettuata rispetto al sistema di riferimento GPS, calcolando latitudine, longitudine e altitudine in ogni momento durante le varie sperimentazioni. Questa modalità di localizzazione avviene fondamentalmente elaborando le misure fornite dal sensore GPS, filtrando i dati ed eventualmente fondendo i dati forniti dalla IMU.

È stata anche prevista la possibilità di combinare più algoritmi di localizzazione tramite un sistema di fusione sensoriale basato su filtro di Kalman esteso (EKF-Extended Kalman Filter), strategia largamente utilizzata per la localizzazione in robotica mobile [8].

2.1.2 Modulo di pianificazione della traiettoria

Come già accennato nel paragrafo precedente, il modulo di pianificazione della traiettoria si occupa di creare e gestire una lista di posizioni che il veicolo dovrà raggiungere in maniera sequenziale e che rappresentano il percorso all’interno dell’ambiente urbano. Le modalità di pianificazione variano leggermente in base alle modalità di localizzazione (“relativa” o “assoluta”):

- nel caso di localizzazione “relativa” i waypoint vengono forniti attraverso l’interfaccia grafica fornita da ROS (interfaccia chiamata “RVIZ”). L’interfaccia

permette di visualizzare anche la mappa dell'ambiente, gli ostacoli presenti e la traiettoria percorsa e pianificata dal veicolo.

- Nel caso di localizzazione “assoluta”, oltre alla possibilità di utilizzo tramite RVIZ descritta in precedenza, è anche possibile replicare una traiettoria (chiamata anche missione) registrata in precedenza. In questo caso è possibile in una prima fase far navigare il veicolo in modalità manuale (tramite controllo fornito con il radiocomando da un operatore) registrando tutte le misure di posizione fornite dal GPS-RTK. Successivamente, tali posizioni, eventualmente sotto-campionate, vengono utilizzate per definire la lista dei waypoint.

Il modulo è in grado di controllare il momento in cui il veicolo raggiunge un determinato waypoint, comunicando il waypoint successivo al sistema di controllo della traiettoria.

2.1.3 Modulo di generazione e controllo della traiettoria

Il modulo di controllo della traiettoria è predisposto per ricevere i waypoint dal sistema di pianificazione e le posizioni del veicolo dal modulo di localizzazione per generare in ogni momento la traiettoria migliore dalla posizione attuale del veicolo al waypoint specificato. La traiettoria inoltre tiene conto sia degli ostacoli presenti (la cui posizione viene fornita dal modulo di rilevazione degli ostacoli), sia delle proprietà strutturali e della cinematica del veicolo (in particolare il tipo di controllo, es: differential drive, sterzo di ackermann, ecc....). La traiettoria così generata viene tradotta in segnali di controllo “di basso livello” da fornire al veicolo. I segnali di controllo sono rappresentati da setpoint di velocità di avanzamento e di rotazione (in caso di sterzata) che, inviati al nodo ROS di interfaccia con la centralina del veicolo, vengono tradotti in segnali di riferimento comunicati agli inverter che controllano i motori elettrici connessi a ciascuna ruota. Il modulo di controllo della traiettoria che è stato testato in campo è basato sulla strategia Time-Elastic Band (TEB) [9], ed è in grado di generare delle traiettorie che tengano conto della cinematica di un veicolo dotato di sterzo di Ackermann (quindi con raggio di curvatura minimo maggiore di zero). Benché il veicolo elettrico sviluppato abbia quattro ruote sterzanti e quindi sia in grado di ruotare con raggi di curvatura anche molto stretti, traiettorie con cinematica di tipo “sterzo di Ackermann” risultano più semplici da controllare, con sforzo minore da parte dei motori.

2.1.4 Modulo di rilevazione degli ostacoli

Questo modulo ha lo scopo di rilevare gli ostacoli fissi ed in movimento che il veicolo si trova ad incontrare, in modo tale che la traiettoria possa essere modificata allo scopo di evitare ogni possibile collisione. Il sistema effettua l'analisi delle scansioni laser fornite dal sensore LIDAR costruendo due mappe che contengono le indicazioni degli ostacoli presenti. Le due mappe sono rispettivamente:

- **mappa di occupazione locale:** contiene gli ostacoli, fissi e mobili, che sono attualmente visibili dal veicolo. Tale mappa è rappresentata da una griglia planare (coast-map) in cui ogni cella rappresenta una piccola area dell'ambiente di lavoro del veicolo. Ogni cella può essere libera, se è attraversabile, oppure occupata, se è presente un ostacolo.
- **mappa di occupazione globale:** anch'essa rappresentata da una griglia di occupazione (coast-map), riporta tutti gli ostacoli fissi che sono stati rilevati durante la navigazione, anche quelli non più visibili.

Le posizioni degli ostacoli rilevati da questo sottosistema vengono comunicate al modulo di controllo della traiettoria, che modificherà localmente il percorso affinché ogni movimento del veicolo sia libero da collisioni. In figura 4 viene mostrata la mappa di occupazione locale calcolata a partire dai dati raccolti dal sensore LIDAR sulla sequenza ENEA.

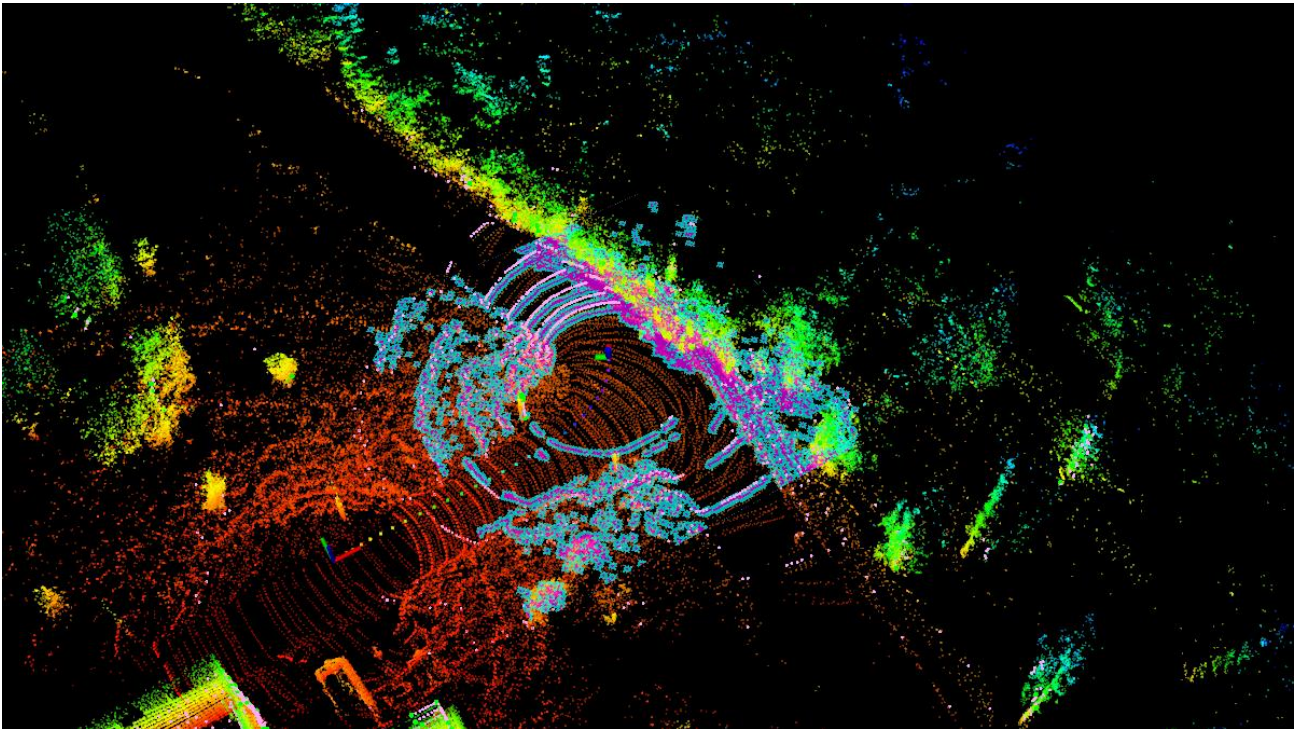


Figura 4: mappa ottenuta con il metodo SC-LEGO-LOAM e mappa di occupazione locale (rappresentate dalle macchie azzurro/viola)

2.1.5 Modulo di conversione di basso livello

L'ultimo modulo che compone il sistema di navigazione autonoma si occupa di calcolare i segnali di basso livello da trasmettere alle schede di controllo collegati direttamente ai motori elettrici. Nonostante il veicolo elettrico possieda quattro ruote sterzanti, durante gli esperimenti di guida si prevede di non sfruttare lo sterzo delle ruote posteriori, rendendo la cinematica della piattaforma quella classica delle automobili, caratterizzata dallo sterzo di Ackermann. Questo tipo di cinematica è caratterizzata da un moto definito attraverso due parametri, la velocità di avanzamento e l'angolo di sterzo (o raggio di curvatura). Per questo motivo il nodo di conversione di basso livello si occupa di ricevere i segnali restituiti dal modulo di controllo della traiettoria, costituiti da valori di velocità lineari ed angolari e di convertirli in valori di velocità di avanzamento ed in set-point per l'angolo di sterzata del veicolo (o di raggio di curvatura). I comandi così convertiti possono essere inviati all'elettronica di basso livello per trasformarli in segnali elettrici da trasmettere ai motori e servomotori elettrici.

3 Esperimenti su sequenze raccolte dal veicolo in ambiente urbano

Sono stati raccolti e forniti dal gruppo di ricerca del centro ENEA sequenze di dati raccolti guidando il veicolo elettrico su strada. Il veicolo è stato fatto girare in guida manuale su strada registrando una file rosbag utilizzando il framework ROS [10] e contenente tutte le misure fornite dai sensori. I dati contenuti all'interno della sequenza comprendono: le immagini raccolte dalla stereocamera anteriore, le scansioni fornite dal Velodyne VLP16, i valori di misurazione inerziale calcolate dalla IMU e valori di posizionamento estratte dal modulo GPS. Come già accennato nella sezione introduttiva di questo documento, una singola misura proveniente da un sensore viene codificato secondo un formato standardizzato all'interno del framework di ROS andando a formare un "messaggio ROS" pronto per essere condiviso con il resto del sistema software tramite delle entità chiamate **topics**. Il contenuto del messaggio ROS dipende dalla tipologia di sensore che trasmette i dati: può, ad esempio, contenere la scansione 3-D di un ambiente in uno specifico istante effettuata da un lidar, le coordinate geografiche del veicolo calcolate dal modulo GPS, un'immagine catturata da una telecamera, o gli angoli di orientamento stimati dalla IMU. Nella tabella 2 vengono mostrate alcune caratteristiche dei dati analizzati: la sequenza è costituita da 3755 immagini stereo, 2481 messaggi ROS provenienti dal sensore LIDAR, 501 messaggi ROS provenienti dal sensore GPS e 2505 provenienti dal sensore IMU, per una durata complessiva di 4 minuti e 10 secondi.

Parametro	Valore
Durata della sequenza	4:10 s
numero di campioni GPS	501
numero di campioni IMU	2505
numero di campioni LIDAR	2481
numero di immagini stereo	3755

Tabella 2: caratteristiche della sequenza raccolta in strada



Figura 5: Immagini estratte dalla sequenza analizzata.

In figura 5 sono mostrati alcuni fotogrammi della sequenza analizzata durante i test. La sequenza raccolta è stata processata con gli algoritmi di localizzazione introdotti precedentemente (anche essi rappresentati da nodi ROS) e sono stati confrontati i risultati ottenuti dalle varie strategie con il ground truth rappresentato dalle misure GPS. Su questa sequenza sono stati testati due algoritmi basati su sistemi di visione, ossia il metodo di visual SLAM ORB-SLAM3 ed il metodo di visual odometry DSO, ed il metodo SLAM basato sull'utilizzo del LIDAR Velodyne VLP 16 LEGO-LOAM. Nella tabella 3 viene mostrato l'errore traslazionale (l'errore nella stima della posizione spaziale fornita dal metodo e confrontando i metodi con il ground truth, che in questi esperimenti è rappresentato dalle misure di posizionamento fornite dal GPS) ottenuto dai tre metodi. L'errore è rappresentato come RMSE (*Root Mean Squared Error*), ossia come errore quadratico medio.

Metodo	RMSE traslazionale
ORB-SLAM 3	1.507339 [m]
DSO	2.267278 [m]
LEGO-LOAM	1.329090 [m]

Tabella 3: performance ottenute dai metodi di localizzazione sulle sequenze raccolte su strada ed espresse come errore quadratico medio traslazionale.

Dai risultati è possibile notare che DSO ottiene le performance peggiori, con il valore di RMSE più alto tra i tre metodi analizzati. I due metodi di SLAM ottengono risultati migliori rispetto al metodo di visual odometry, con il metodo LEGO-LOAM, basato sull'utilizzo del LIDAR, che risulta il migliore tra i tre. Un risultato di questo tipo era atteso, poiché è noto che i metodi di localizzazione basati su visione sono influenzati da fenomeni, come variazioni di illuminazione o di appearance, che possono causare

errori di stima, mentre sensori di distanza come laser e LIDAR, risultano molto più robusti e precisi.

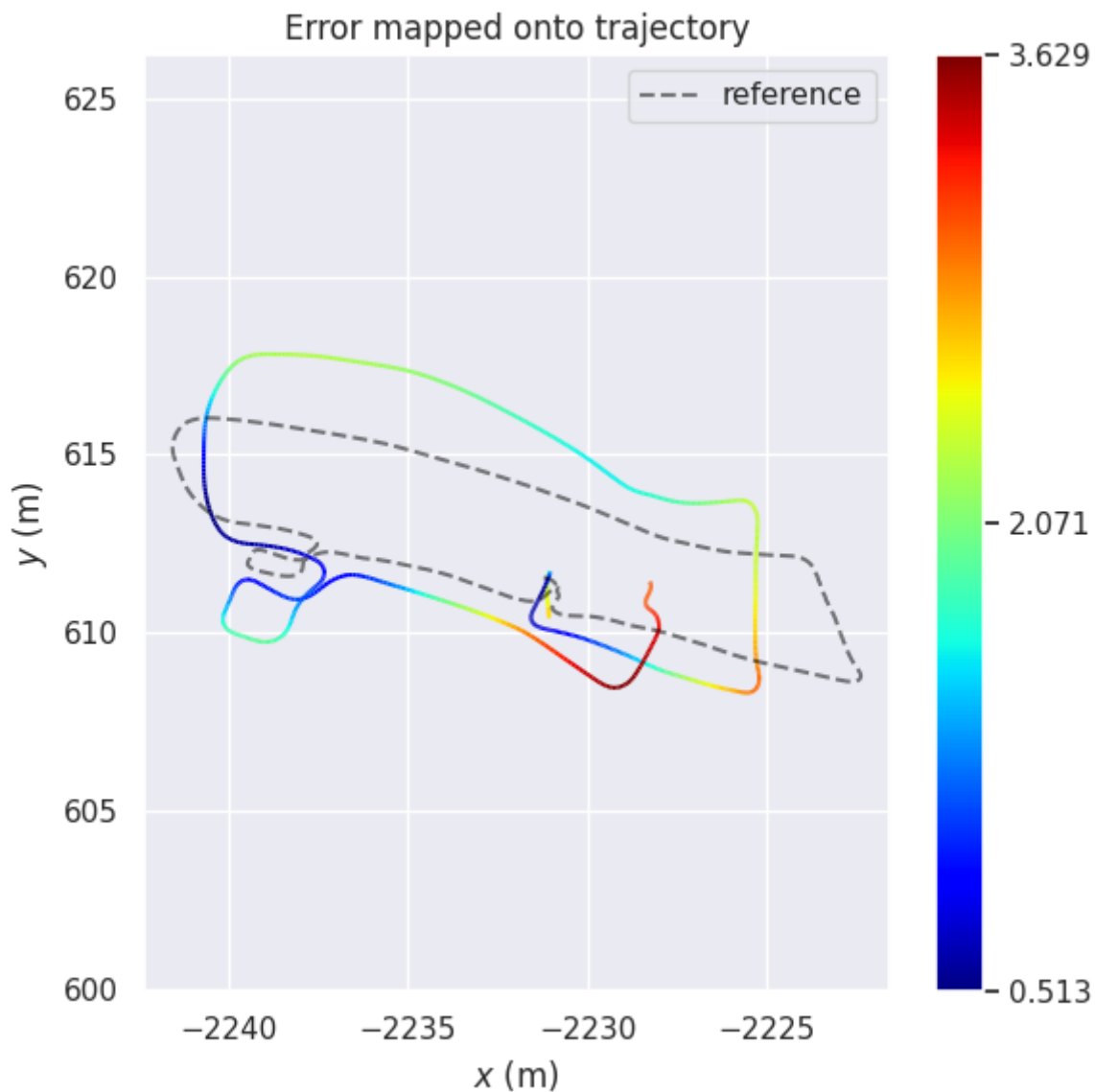


Figura 6: confronto tra ground truth (rappresentata dalle posizioni fornite dal GPS e mostrata dalla linea tratteggiata) e traiettoria stimata dal metodo DSO (rappresentata dalla linea colorata). I colori inoltre codificano l'errore in quella parte della traiettoria rispetto al ground truth

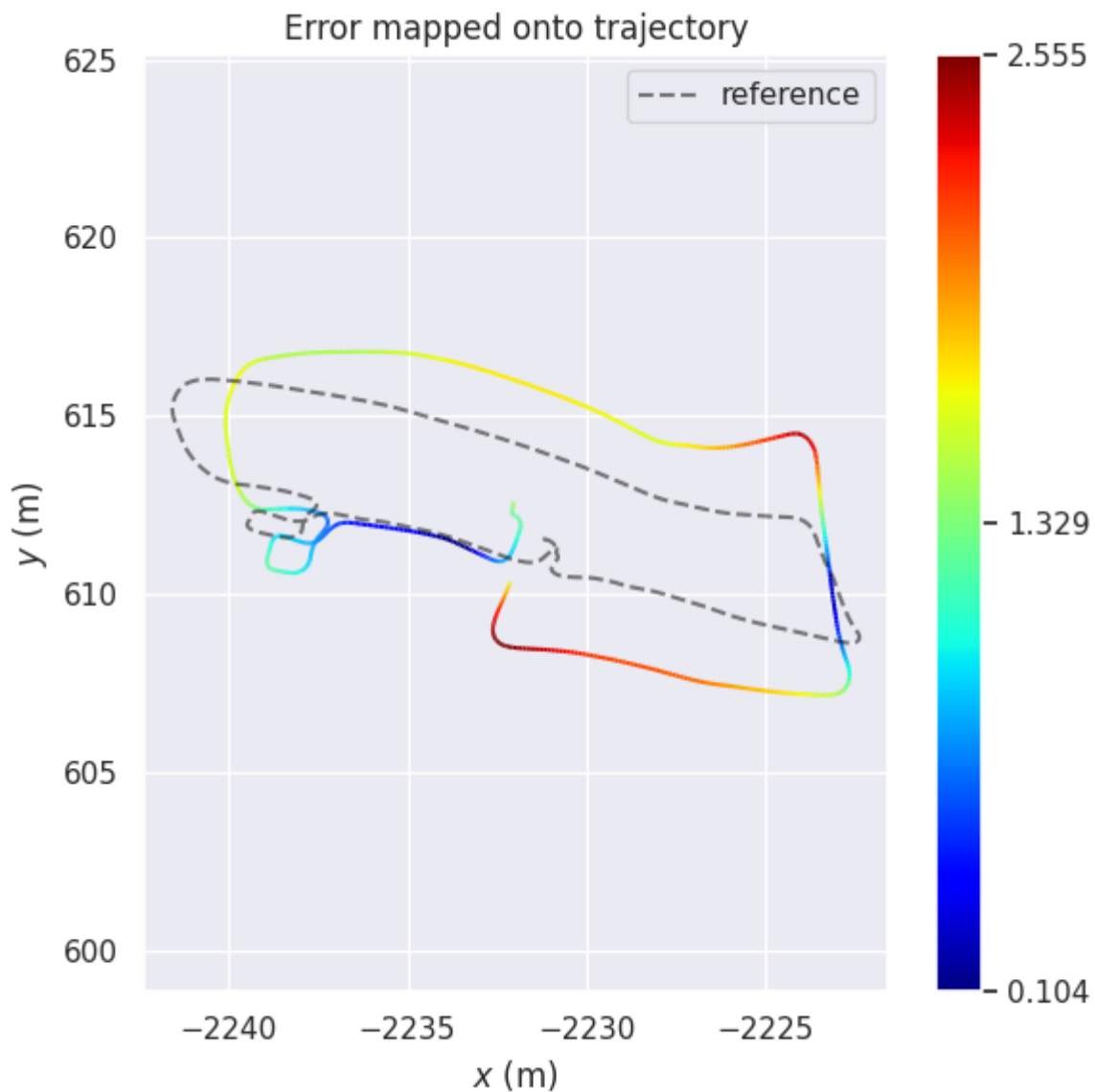


Figura 7: confronto tra ground truth (rappresentata dalle posizioni fornite dal GPS e mostrata dalla linea tratteggiata) e traiettoria stimata dal metodo ORB-SLAM 3 (rappresentata dalla linea colorata). I colori inoltre codificano l'errore in quella parte della traiettoria rispetto al ground truth

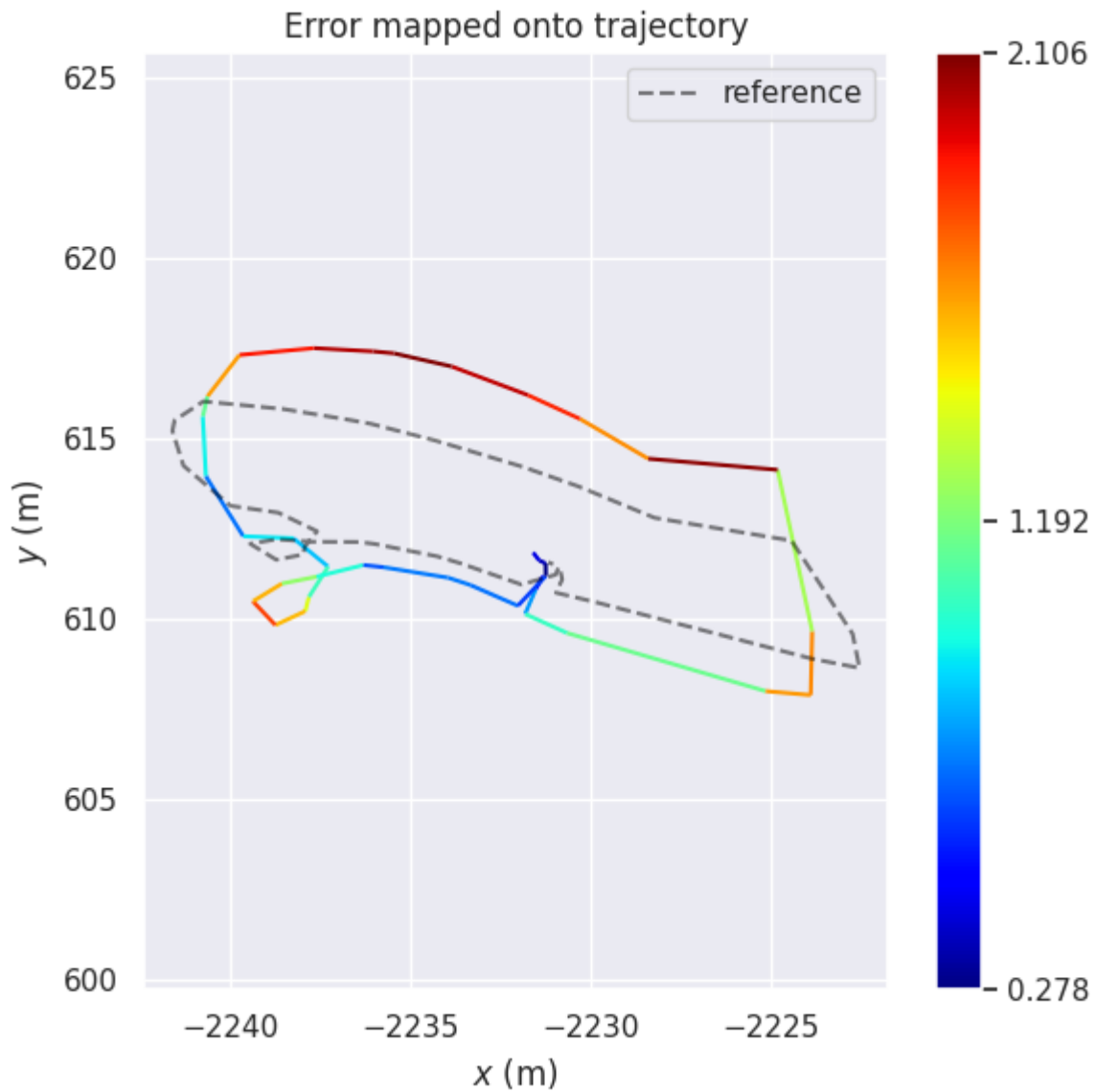


Figura 8: confronto tra ground truth (rappresentata dalle posizioni fornite dal GPS e mostrata dalla linea tratteggiata) e traiettoria stimata dal metodo LEGO-LOAM (rappresentata dalla linea colorata). I colori inoltre codificano l'errore in quella parte della traiettoria rispetto al ground truth

Nelle figure 6-7-8 sono riportate le traiettorie calcolate dai tre metodi, confrontate con la traiettoria di ground truth stimata dal sensore GPS. La traiettoria del GPS è riportata tramite una linea tratteggiata, mentre quelle stimate dai metodi come linea continua colorata, in cui a ciascun colore corrisponde un valore di errore per quella porzione di traiettoria rispetto al ground truth.

4 Esperimenti di guida autonoma in ambiente urbano

Il sistema di localizzazione e navigazione descritto nel capitolo 2 è stato validato anche tramite esperimenti di guida autonoma condotti con il veicolo elettrico in ambiente urbano. Più precisamente, sono stati condotti dei test sulle strade presenti negli spazi del centro di ricerca ENEA di Casaccia. Gli esperimenti sono stati effettuati installando ed eseguendo il sistema software di navigazione all'interno del computer di bordo presente nel veicolo elettrico. Per interagire da remoto con il veicolo è stato inoltre predisposta una control station costituita da un laptop.

Sono stati condotti due tipi di esperimenti di guida autonoma: nella prima tipologia di esperimenti il veicolo è stato fatto navigare in un ambiente non noto a priori in modalità "SLAM", quindi in mapping e localizzazione simultanea. Come anche evidenziato nel capitolo precedente, LEGO-LOAM è l'approccio che garantisce la maggior precisione di localizzazione e, di conseguenza, è stato scelto come metodo di localizzazione anche per gli esperimenti in tempo reale. Per la visualizzazione e l'interazione con il sistema di navigazione è stato utilizzato il tool di RVIZ di ROS, che fornisce la mappa in tempo reale dell'ambiente e consente all'utente di impostare i waypoints che il veicolo deve raggiungere. Una volta ricevuti i waypoints, i moduli di pianificazione e generazione della traiettoria calcoleranno i comandi da inviare al controllore di basso livello.

Nella seconda tipologia di esperimenti il sistema di navigazione è stato fatto funzionare in modalità "localizzazione", organizzando l'esperimento in due fasi:

1. Fase 1 - "Mapping": viene eseguito il mapping "preliminare" dell'ambiente guidando manualmente il veicolo elettrico all'interno delle strade del centro di ricerca ENEA ed utilizzando una versione di LEGO-LOAM che consente il salvataggio della mappa stimata.
2. Fase 2 - "Localizzazione": utilizzando la mappa precedentemente salvata viene avviata la fase di localizzazione in ambiente noto. La modalità di "localizzazione" è stata implementata con una versione modificata di LEGO-LOAM in grado di utilizzare una mappa dell'ambiente pre-calcolata e di stimare la posizione del veicolo utilizzando le misurazioni del LIDAR e della IMU in tempo reale. L'interazione con il sistema di pianificazione e navigazione viene condotta analogamente alla prima tipologia di esperimenti: usando RVIZ si può osservare la mappa dell'ambiente e posizionare i waypoints da raggiungere.

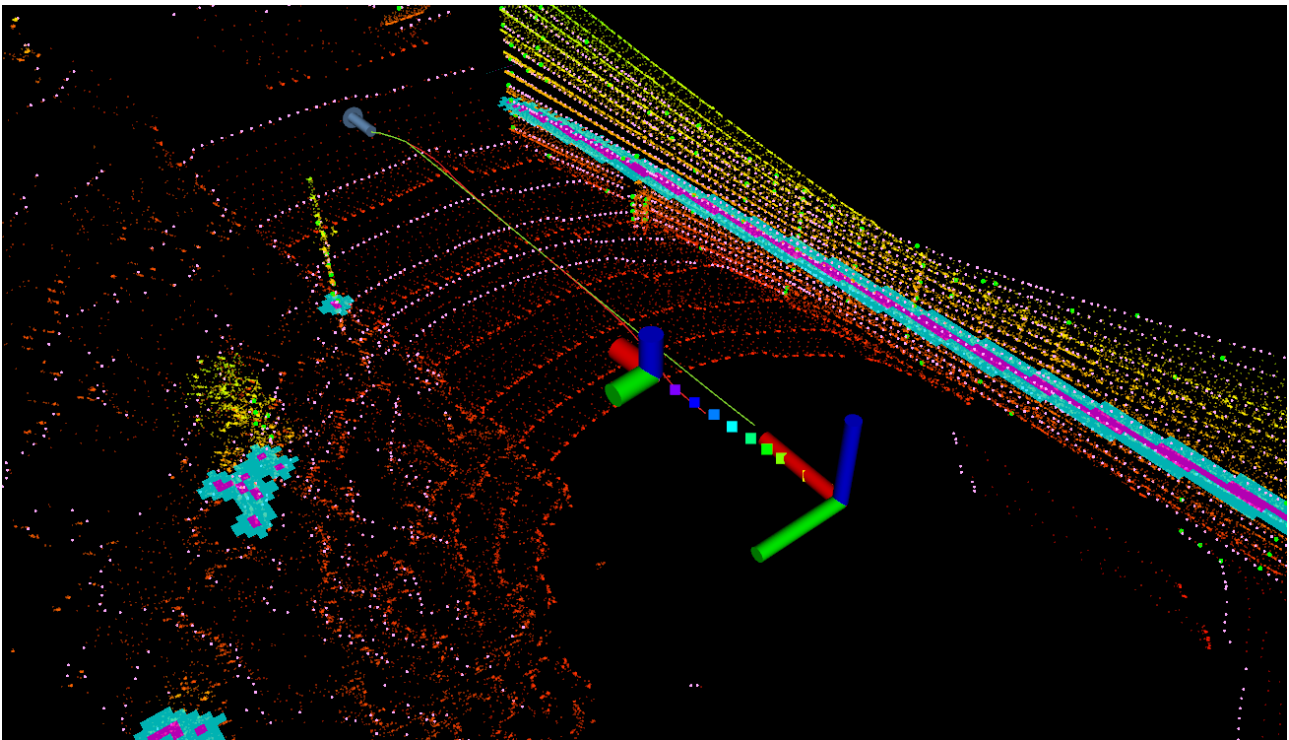


Figura 9: Esperimento di guida autonoma con il veicolo in ambiente urbano. In particolare, viene mostrata la mappa 3D ottenuta dal metodo LEGO-LOAM, la mappa locale degli ostacoli (punti azzurro/viola), il punto di partenza (rappresentata dai tre assi verde-rosso-blu grandi), il punto di arrivo (freccia blu), la posizione attuale del veicolo (tre assi verde-rosso blu piccoli), la traiettoria percorsa dal veicolo (serie di punti colorati), la traiettoria calcolata dal global planner (linea verde) e dal local planner (linea rossa).

Le traiettorie pianificate dal sistema di navigazione vengono modificate in caso di presenza di ostacoli al fine di evitare collisioni, mentre il nodo di conversione di basso livello si occuperà di trasformare i comandi di velocità lineare ed angolare in valori di velocità di avanzamento e di angolo di sterzata.

La figura 9 mostra un esperimento di guida autonoma con il veicolo in strada, mentre in figura 10 viene mostrato il veicolo elettrico.

L'uso del sistema di navigazione in ambiente non noto (modalità "SLAM") permette l'utilizzo degli algoritmi senza la necessità di scansioni preliminari; tuttavia, la scelta dei waypoints risulta più complessa in quanto parte dell'ambiente non risulterà visibile. Al contrario l'uso del sistema in modalità "localizzazione" permette di sfruttare la mappa calcolata in precedenza anche per una scelta più precisa dei waypoints.



Figura 10: veicolo elettrico durante gli esperimenti di guida autonoma in ambiente urbano.

5 Conclusioni

In conclusione, in questa ultima fase del progetto svolta nell'arco del 2021, è stato implementato e sperimentato il sistema di localizzazione e guida autonoma per il veicolo elettrico. Il sistema è composto da un insieme di moduli funzionali, in particolare: il modulo di localizzazione del veicolo, il modulo di pianificazione della traiettoria, il modulo di generazione e controllo dei comandi, il modulo di rilevazione degli ostacoli ed il modulo di controllo di basso livello. Gli esperimenti sono stati condotti grazie alla collaborazione con il gruppo di ricerca del laboratorio di Robotica ed Intelligenza Artificiale del Centro di Ricerca di Casaccia, sia raccogliendo sequenze di dati con il veicolo elettrico e processandoli off-line, sia effettuando dei test di navigazione autonoma in tempo reale in ambiente urbano. Confrontando i diversi metodi SLAM e di visual odometry per la localizzazione, è emerso che gli approcci basati su LIDAR garantiscono una maggiore precisione rispetto a quelli basati su visione. I test di navigazione autonoma in tempo reale, infine, hanno dimostrato l'efficacia del sistema a seguito dell'integrazione con il veicolo elettrico.

6 Riferimenti bibliografici

1. Wang, Han, et al. "F-loam: Fast lidar odometry and mapping." *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.
2. Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE.
3. Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2.
4. Kim, Giseop, and Ayoung Kim. "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map." *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
5. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Daniela, R. (2020). Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE.
6. Campos, C., Elvira, R., Gomez, J. J., Montiel, J. M. M., and Tardos, J. D. (2020). ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv preprint arXiv:2007.11898*.
7. Engel, J., Koltun, V., and Cremers, D. (2016). Direct sparse odometry. *CoRR*, abs/1607.02565. Engel, J., Schoeps, T., and Cremers, D. (2014). Lsd-slam: large-scale direct monocular slam. volume 8690, pages 1–16.
8. Moore, Thomas, and Daniel Stouch. "A generalized extended kalman filter implementation for the robot operating system." *Intelligent autonomous systems 13*. Springer, Cham, 2016. 335-348.
9. C. Rösmann, F. Hoffmann and T. Bertram: Kinodynamic Trajectory Optimization and Control for Car-Like Robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, Sept. 2017.
10. Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." *ICRA workshop on open source software*. Vol. 3. No. 3.2. 2009.

7 Abbreviazioni ed Acronimi

BRIEF: Features from Accelerated Segment Test

EKF: Extended Kalman Filter

FAST: Features from Accelerated Segment Test

FLOAM: Fast LOAM

GPS: Global Positioning System

IMU: Inertial Measurement Unit

LEGO-LOAM: Lightweight and Ground-Optimized LOAM

LIDAR: Laser Imaging Detection and Ranging

LIO-SAM: Lidar Inertial Odometry via Smoothing And Mapping

LOAM: LIDAR Odometry and Mapping

ORB: Oriented FAST and Rotated BRIEF

SBC: Single Board Computer

SLAM: Simultaneous Localization and Mapping

RTK: Real-Time Kinematic

TEB: Time Elastic Band

8 Breve Curriculum Vitae del Gruppo di Lavoro del Dipartimento di Ingegneria, Università degli Studi di Perugia

Enrico Bellocchio ha conseguito la Laurea Magistrale in Ingegneria dell'Informazione e dell'Automazione presso l'Università degli Studi di Perugia nel 2014. Dal 2014 fa parte dell'*Intelligent Systems, Automation and Robotics Laboratory* (ISARLab) come assegnista di ricerca. La sua attuale attività di ricerca è focalizzata su machine learning, computer vision e robotica.

Mirko Leomanni ha conseguito il Dottorato di Ricerca in Ingegneria dell'Informazione presso l'Università degli Studi di Siena nel 2015. Dal 2015 al 2021 ha fatto parte del Gruppo di Controlli Automatici del Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche dell'Università degli Studi di Siena. Attualmente è ricercatore presso l'*Intelligent Systems, Automation and Robotics Laboratory* (ISARLab) dell'Università degli Studi di Perugia. I suoi interessi di ricerca includono i sistemi di controllo aerospaziali, l'ottimizzazione e le tecniche di navigazione autonoma.

Gabriele Costante ha conseguito il Dottorato di Ricerca in Ingegneria dell'Informazione presso il Dipartimento di Ingegneria dell'Università degli Studi Perugia nel 2016. Attualmente è Ricercatore presso l'*Intelligent Systems, Automation and Robotics Laboratory* (ISARLab) e docente dei corsi di *Computer Vision and Robot Perception* e *Machine Learning and data Analysis* presso il Dipartimento di Ingegneria dell'Università degli Studi di Perugia. I suoi interessi di ricerca includono l'intelligenza artificiale, la robotica, la visione artificiale e il Machine Learning.

Mario Luca Fravolini ha conseguito il dottorato di ricerca in Ingegneria Elettronica presso l'Università degli Studi di Perugia nel 2000. Ha lavorato come assistente di ricerca nel gruppo di teoria dei controlli presso la School of Aerospace Engineering, Georgia Institute of Technology, e presso il Dipartimento di Ingegneria Meccanica e Aerospaziale della West Virginia University. Attualmente è professore associato presso il Dipartimento di Ingegneria dell'Università degli Studi di Perugia. I suoi interessi di ricerca includono: Diagnosi dei guasti, Controllo intelligente e Controllo Intelligente, e Adattativo e Imaging Biomedico.

Paolo Valigi ha conseguito il dottorato di ricerca presso l'Università di Roma "Tor Vergata" nel 1991. Dal 1990 al 1994 ha collaborato con la Fondazione Ugo Bordoni. Dal 2004 è Professore Ordinario presso l'Università degli Studi di Perugia, Dipartimento di Ingegneria. Attualmente è responsabile dell'*Intelligent Systems, Automation and Robotics Laboratory* (ISARLab). I suoi interessi di ricerca includono la teoria e applicazioni del controllo automatico, l'identificazione dei sistemi, la robotica e la System Biology.

Lista delle pubblicazioni recenti più rilevanti

- Devo, A.; Dionigi, A.; Costante, G. “Enhancing continuous control of mobile robots for end-to-end visual active tracking”, *Robotics and Autonomous Systems*, 2021.
- Cartocci, N.; Napolitano, M. R.; Costante, G.; Fravolini, M. L. “A Comprehensive Case Study of Data-Driven Methods for Robust Aircraft Sensor Fault Isolation”, *Sensors*, 2021.
- Cascianelli, S.; Costante, G.; Crocetti, F.; Ricci, E.; Valigi, P.; Fravolini, M. L. “Data-based design of robust fault detection and isolation residuals via LASSO optimization and Bayesian filtering” *Asian Journal of Control*, 2021.
- Brilli R., Pozzi M., Giorgetti F., Fravolini M. L., Valigi P., Prattichizzo P., Costante G., “Monocular Reactive Collision Avoidance Based on Force Fields for Enhancing the Teleoperation of MAVs” 20th International Conference on Advanced Robotics - Dicembre 6-10, 2021. Ljubljana, Slovenia
- Cartocci N., Crocetti F., Costante G., Valigi P., Napolitano M., Fravolini M. L., “Data-Driven Sensor Fault Diagnosis Based on Nonlinear Additive Models and Local Fault Sensitivity” 20th International Conference on Advanced Robotics - dicembre 6-10, 2021. Ljubljana, Slovenia
- Felicioni S., Legittimo M., Fravolini M. L., Costante G., “GOLN: Graph Object-Based Localization Network” 20th International Conference on Advanced Robotics - dicembre 6-10, 2021. Ljubljana, Slovenia
- Crocetti, F.; Costante, G.; Fravolini, M. L.; Valigi, P., “Tire-road friction estimation and uncertainty assessment to improve electric aircraft braking system” *IEEE 28th Mediterranean Conference on Control and Automation (MED)*, 2021.
- Cartocci, N.; Napolitano, M.R.; Costante, G.; Crocetti, F.; Valigi, P.; Fravolini, M. L. “A Robust Data-Driven Fault Diagnosis scheme based on Recursive Dempster–Shafer Combination Rule” *IEEE 28th Mediterranean Conference on Control and Automation (MED)*, 2021.
- Costante, G., Mancini, M. “Uncertainty Estimation for Data-driven Visual Odometry”. *IEEE Transactions on Robotics*, 2020.
- Devo, A., Mezzetti, G., Costante, G., Fravolini, M. L., & Valigi, P. “Towards Generalization in Target-Driven Visual Navigation by Using Deep Reinforcement Learning”. *IEEE Transactions on Robotics*, 2020.
- Fravolini, M. L., Costante, G., Yucelen, T., & Napolitano, M. R. “Interval Prediction Models for Data-Driven Design of Aerial Vehicle’s Robust Adaptive Controllers”. *Journal of Guidance, Control, and Dynamics*, 1-17, 2020.
- Crocetti F., Costante G., Fravolini M.L., Valigi P. “A Data-Driven Slip Estimation Approach for Effective Braking Control under Varying Road Conditions” 2020 28th Mediterranean Conference on Control and Automation (MED), 496-501
- Cartocci N., Costante G., Napolitano M.R., Valigi P., Crocetti F., Fravolini M.L. “PCA Methods and Evidence Based Filtering for Robust Aircraft Sensor Fault Diagnosis” 2020 28th Mediterranean Conference on Control and Automation (MED), 550-555

- Cascianelli S., Costante G., Crocetti F., Ricci E., Valigi P., Fravolini M.L. "Data-based design of robust fault detection and isolation residuals via LASSO optimization and Bayesian filtering" Asian Journal of Control, 2020
- Devo, Alessandro, Costante G., and Paolo Valigi. "Deep Reinforcement Learning for Instruction Following Visual Navigation in 3D Maze-Like Environments." IEEE Robotics and Automation Letters 5(2): 1175-1182, 2020.
- Costante G., Ciarfuglia T. A. "Ls-vo: Learning dense optical subspace for robust visual odometry estimation", IEEE Robotics and Automation Letters, 2018.
- Mancini M., Costante G., Valigi P., Ciarfuglia T. A. "J-MOD2: Joint Monocular Obstacle Detection and Depth Estimation", IEEE Robotics and Automation Letters, 2018.
- Abdollahyan M., Cascianelli S., Bellocchio E., Costante G., Ciarfuglia T.A., Bianconi F., Smeraldi F., Fravolini M.L. "Visual Localization in the Presence of Appearance Changes Using the Partial Order Kernel", 26th European Signal Processing Conference (EUSIPCO), 2018.
- Cascianelli, S.; Costante, G.; Bellocchio, E.; Valigi, P.; Fravolini, M. L.; Ciarfuglia, T. A. "Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features.", Robotics and Autonomous Systems, 92, 53-65, 2017.
- Costante, G.; Mancini, M.; Valigi, P; Ciarfuglia, T.A. "Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation", IEEE Robotics and Automation Letters, 2016, Best Vision Paper Finalist alla conferenza IEEE International Conference on Robotics and Automation.
- Mancini, M.; Costante, G.; Valigi, P.; Ciarfuglia, T.A. "Fast robust monocular depth estimation for Obstacle Detection with fully convolutional networks", IEEE/RSJ International Conference on Intelligent Robotics and Systems, 2016.
- Leomanni, Mirko, et al. "Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target." Journal of Guidance, Control, and Dynamics (2022): 1-13.
- Leomanni, Mirko, et al. "Sum-Of-Norms MPC for linear periodic systems with application to spacecraft rendezvous." 2020 59th IEEE Conference on Decision and Control (CDC). IEEE, 2020.
- Garulli, Andrea, Antonio Giannitrapani, and Mirko Leomanni. "Minimum switching limit cycle oscillations for systems of coupled double integrators." 53rd IEEE Conference on Decision and Control. IEEE, 2014.